Les descripteurs d'un son Librairie Matlab SPL et fichiers de descriptions ".sig""

Guillaume Lemaître, Emmanuel Gallo

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

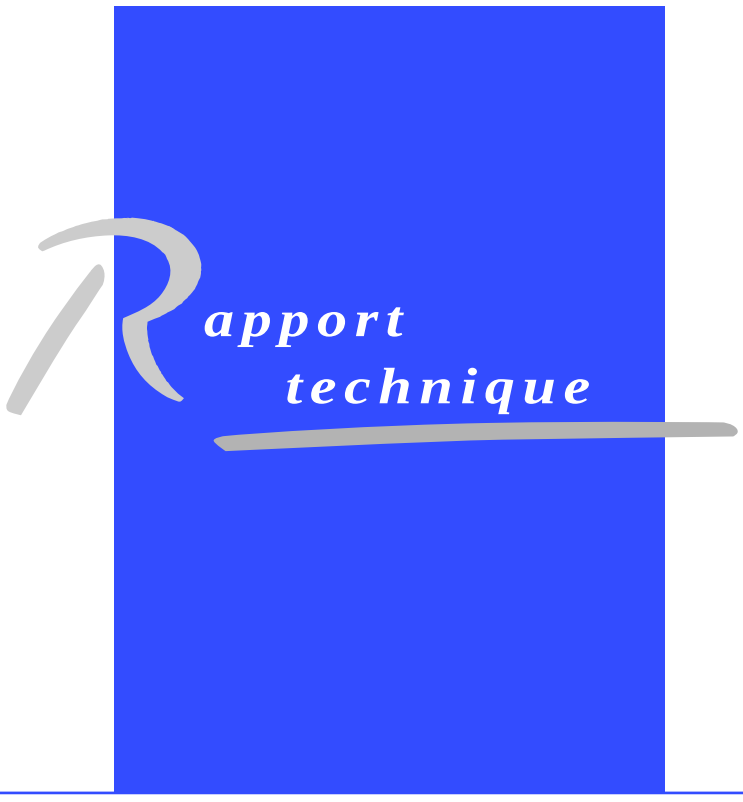
Les descripteurs d'un son
Librairie Matlab SPL et fichiers de descriptions ".sig*"*

Guillaume Lemaitre — Emmanuel Gallo

N° 0317

Janvier 2006

Thème COG



ISSN 0249-0803

Les descripteurs d'un son
Librairie Matlab SPL et fichiers de descriptions ".sig"

Guillaume Lemaitre , Emmanuel Gallo

Thème COG — Systèmes cognitifs
Projets Reves

Rapport technique n° 0317 — Janvier 2006 — 26 pages

Résumé : Le traitement du signal audio rapide travaille avec des descriptions compactes des sons plutôt qu'avec les sons eux-mêmes. Dans une première partie nous introduisons cette notion de description et nous proposons un état de l'art des différents types de description utilisés. Ces descriptions incorporent le plus souvent des quantités telles que l'énergie ou le spectre de puissance à court-terme. Cependant, énergie et puissance sont des concepts qui sont souvent confondus ou mal interprétés, et leur définition dépend des conventions de calcul qui ont été choisies pour les signaux numérisés. Nous rappelons dans une deuxième partie les définitions théoriques (physiques) de ces concepts, et nous proposons nos propres conventions de calcul. Nous définisson alors un ensemble de formules pour le calcul exact (i.e. cohérent avec leur définition physique) de l'énergie, de la puissance ou de la valeur RMS en dB SPL. Ces définitions sont implémentées dans une librairie Matlab (SPL) qui sert à calculer les descripteurs d'un fichier son, et les incorpore dans un fichier de description dont nous avons défini le format (.sig). La dernière partie de ce rapport décrit la syntaxe de ces fonctions. Finalement, la troisième partie de ce rapport décrit la stucture des fichiers .sig.

Mots-clés : Description d'un son, énergie, puissance, Matlab, SPL toolbox, .sig

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Sound description
“Matlab SPL Library” and “.sig” descriptions files

Abstract: Fast audio processing uses compact sound descriptions attached to each file. In a first part, we introduce the concept of sound description and we propose a state-of-the-art of the different kinds of sound descriptions already used. These descriptions always include energy and power short-time power spectra. Energy and power are concepts that are often confused or misinterpreted. Morever, their definition depends on normalization convention. In a second part, we remind the theoretical and physical definitions of these concepts, and we define our own normalization conventions. With these definitions we propose an exact computation standard for energy, power and dB SPL. These definitions are implemented in a Matlab library (SPL) which aims at computing a sound description in a specific file (.sig). In a third part, we present the functions of the SPL library. Finally the third part of the report describes the .sig file structure.

Key-words: Sound descriptionEnery, power, Matlab, SPL toolbox, .sig

1 Description et codage d'un son

Décrire un son consiste à en extraire des informations choisies qui seront nécessaires pour réaliser un certain nombre de calculs. Il s'agit donc d'associer à un fichier son un fichier contenant un nombre limité de *descripteurs*. Il s'agit d'une réduction d'information. Le fichier son du départ contient le signal dans son intégralité (à l'échantillonnage et à la quantification près), alors que le fichier de description ne contient que quelques informations sélectionnées. Il convient cependant de distinguer la *description* d'un son de la *compression* d'un son. Dans le premier cas, l'objectif est de ne retenir que quelques informations nécessaires à la réalisation de calculs dont les applications ont besoin ; dans le second cas, il s'agit de réduire la taille occupée par le fichier son, en faisant en sorte que cette réduction soit perceptivement la plus légère possible pour l'auditeur[1]. C'est pourquoi ce paragraphe ne s'intéressera pas aux techniques de compression audio telles que celles basées sur le masquage auditif (MPEG-1 layer 3, etc. [18]).
Le *codage* de la description d'un son est la façon dont est organisé le fichier qui contient les descripteurs. Ce codage doit être optimisé pour permettre des opérations rapides sur la descriptions (balayage, calculs, etc.).

Il existe trois grandes familles d'informations pouvant être calculés à partir d'un fichier son :
- les descripteurs psychoacoustiques,
- les représentations basées sur un modèle paramétrique du son,
- les paramètres statistiques du signal.

En fonction du contexte, on parle de caractéristiques, de paramètres, ou de descripteurs du signal. Dans la suite de ce document, ces informations étant calculées dans le but de décrire un signal, on parlera simplement de descripteurs.
Parmi ces différentes types de description, il faut distinguer les représentations inversibles et les représentations desrcriptives. Les premières permettent de reconstruire le signal original, alors que les dernières ne le permettent pas.
Les descripteurs statistiques sont cités et détaillés au paragraphe 1.1, certains descripteurs liés à une modélisation paramétrique du son sont détaillés au pagraphe 1.2. Nous ne détaillons pas les descripteurs psychoacoustiques ici, mais des détails peuvent être trouvés dans les ouvrages de référence [30, 7, 14], ou dans des publications décrivant leur utilisation pour diverses applications [19, 12, 13]. Il existe plusieurs tentatives pour intégrer ces différentes caractéristiques dans des représentations structurées (*Structured audio* [26], SDIF [28], MPEG-7 [9] [8] [19]).

1.1 Les paramètres statistiques du signal

Il existe un multitude de manières de stocker des informations statistiques sur les signaux. À titre d'exemple, on peut citer :
- Les coefficients d'autocorrélation du signal [22, 20]
- Le taux de passage du signal par zéro [23, 29, 22, 20]
- La moyenne, la variance du signal au cours du temps [27]
- Les fluctuations temporelles de l'enveloppe temporelle, le temps d'attaque, de décroissance de l'enveloppe temporelle [10, 22, 20]
- La largeur de la distribution de l'amplitude du signal [1]
- Le centre de gravité spectral et ses variations temporelles [1, 22, 20]
- La tonalité [1]
- La transformée de Fourier à court terme (à la base de la plupart des autres descripteurs spectraux)
- L'énergie dans chaque trame [23, 29, 22, 20]
- Les fluctuations temporelles de l'enveloppe spectrale (delta cepstrum), les paramètres statistiques de l'enveloppe spectrale (étalement, skewness, kurtosis [15, 22, 20]
- La fréquence fondamentale et ses variations au cours du temps [29, 1]
- Le trajet des partiels [29]
- La fonction d'autocorrélation de l'énergie [27]

Parmi ces descriptions, nous allons nous attarder sur les coefficients cepstraux. Ces coefficients servent à coder de manière progressive l'enveloppe spectrale du signal. Ils sont très courramment utilisés.

Les coefficients cepstraux Les coefficients cepstraux sont très utilisés dans la communauté de la *Computational Auditory Scene Analysis* [2]), et sont issus du traitement de la parole. Dans ce paragraphe le principe général du calcul du cepstre est d'abord détaillé, puis celui des coefficients cepstraux basés sur une échelle de mel.
Le cepstre est une représentation de l'amplitude du spectre du signal. L'idée générale est de décomposer l'amplitude du

[1] Certaines techniques de compression suppriment des informations inaudibles. Dans les deux cas, il s'agit donc de réduire l'information stockée. La différence tient à l'objectif de la réduction d'information

spectre sur une base de fonctions orthogonales (transformée de Fourier ou transformée en cosinus [17]), et d'obtenir ainsi une représentation dont les coefficients sont décorrélés entre-eux, et ordonnés suivant leur importance. Il s'agit, grossièrement parlant, du "spectre du spectre" (par exemple, si le spectre d'amplitude possède une enveloppe périodique, le cepstre va présenter des raies). Le cepstre est censé être une approximation d'une analyse en composantes principales du spectre. Ceci permet, si besoin est de se limiter aux quelques premiers coefficients cepstraux pour compacter la représentation. Le résulat est donc un ensemble de coefficients cepstraux, définis chacun pour une fréquence de la transformée du spectre (*quefrency*, mais on parle plutôt de l'indice du coefficient cepstral). Toute l'information est censée être concentrée dans les premiers coefficients.

Coefficients cepstraux en bandes de mel (mfcc) En tant que tel, le cepstre est donc une transformation du signal, qui n'est pas très compacte, et qui est calculée sur toute la durée du signal.
Pour calculer une représentation du spectre qui soit beaucoup plus compacte, et qui soit exprimée en fonction du temps, une version modifiée du cepstre est introduite : les *mel frequency cepstral coefficients* (mfcc). L'idée est de moyenner le spectre dans des bandes de fréquence correspondant grossièrement au filtrage effectué par la membrane basilaire avant d'effectuer la transormée en cosinus. On réduite de cette manière considérablement le nombre de coefficient cepstraux. Le banc de filtre utilisé utilise l'échelle des mels introduite par Stevens et Volkmann en 1940 (cités par Rabiner [24]). Les filtres sont triangulaires.
La procédure ce calcul des mfcc est donc la suivante (voir la figure 10) :

- Pré-filtrage du signal (passe-haut).
- Fenêtrage (Hamming) du signal (N0 points) en K fenêtres temporelles de N points chacune.
- Calcul de la DFT dans chaque frame (N points), puis de son module (ou du carré du module).
- Filtrage par les filtres triangulaires en échelle mel (en fait, l'amplitude du spectre est pondérée et sommée dans des fenêtres triangulaire), puis passage au logarithme. Typiquement, 40 bandes de mel, de 133 Hz à 6kHz (ce qui revient à déjà filtrer le signal original !)
- Calcul de la transformée en cosinus, (ou de la transformée en cosinus inverse, ou de la transformée de Fourier) pour chaque fenêtre. On obtient donc K * 40 coefficients. Typiquement seuls les 13 premiers coefficients sont conservés.

On se retrouve donc avec une représentation temps-coefficients cepstraux : une matrice dont les colonnes correspondent aux différents pas temporels, et dont les lignes correspondent aux coefficients cepstraux.

Remarques : Cette technique s'inspire des modèles psychoacoustiques d'audition : d'abord par l'echelle mel et les filtres triangulaires qui sont une approximation du filtrage de la membrane basilaire, et ensuite le fait de prendre le logarithme de la sortie des filtres est une tentative de modéliser la sonie. Une implémentation plus fine pourrait prendre des filtres gammatone, une échelle de bark, est un modèle de sonie en puissance.
Les mfcc codent uniquement le spectre d'amplitude du signal. On perd donc toutes les informations de phase.
DCT est utilisée comme un approximation de la transformée de Karhunen-Loeve qui est elle-même censée être équivalente à une PCA, pour la parole [11]. C'est-à dire que les coefficients cepstraux sont censés être décorrélés les uns des autres, ordonnés les uns les autres. Ce n'est pas tout à fait vrai, mais on peut montrer que les vecteurs de base de la transformée en cosinus sont proches de ceux de la transformée de Karhunen-Loeve pour des signaux de parole et de musique [11], et que l'erreur faite en ne prenant que les premiers coefficients est beaucoup plus faible en prenant une DCT qu'un DFT [17].
Les mfcc sont utilisés pour calculer les similarités à l'intérieur d'un morceau de musique [21], ou pour faire de la reconnaissance autmoatique d'évéments sonores [5], etc.

1.2 Les représentations basées sur un modèle paramétrique du son.

A la différence des descriptions qui associent à un son un ensemble de toutes sortes de descripteurs statistiques ou psychoacoustiques redondants, les descripteurs issus des représentations paramètriques des sons sont basées sur l'existence d'un modèle explicite et préalable du signal sonore. Les descripteurs sont les paramètres du modèle. Les sons sont, comme dans les descriptions précédentes, représentés par leur position dans un espace de descripteurs, mais la différence principale réside dans le fait que l'ensemble des descripteurs permet de reconstruire un son qui est censé être très proche du son original, si ce n'est le son original lui-même. Il s'agit donc d'une représentation inversible du son (si tous les descripteurs sont conservés). La description du son est compacte, parce que l'information est en fait contenue dans le modèle.
L'utilisation d'un modèle préalable du signal sonore signifie également que certaines hypothèse ont été faites quant-à la nature des sons à décrire. Tous les sons ne peuvent être décrits par le même modèle [26].

Les modèles sinusoïdaux Il s'agit des modèles qui considèrent qu'un son est constitué d'une somme de sinusoïdes (partie dite "harmonique" ou "périodique", bien que rien n'oblige les partiels à être harmoniques), ajoutée à un bruit (signal aléatoire). On trouve ce modèle dans le logiciel de synthèse ADDITIVE de l'Ircam [3], et dans le modèle SMS (*spectral modeling synthesis*) développé à l'Institut Universari de l'Audiovisual de l'Universitat Pompeu Fabra de Barcelone (IUA/UPF) [9], et dans les modèles dérivés (e.g. les modèles statistiques [4] [6] du LaBRI de l'Université de Bordeaux I). D'une manière générale, dans le cadre de tels modèles, les sons sont décrits à chaque pas temporel par une matrice contenant la fréquence (ou l'indice), l'amplitude et la phase de chaque partiel, et d'un vecteur contenant les propriétés statistiques de la partie bruitée (voir figure 1) .
La technique pour extraire ces caractéristiques consiste à effectuer un ensemble de transformées de Fourier à court terme,

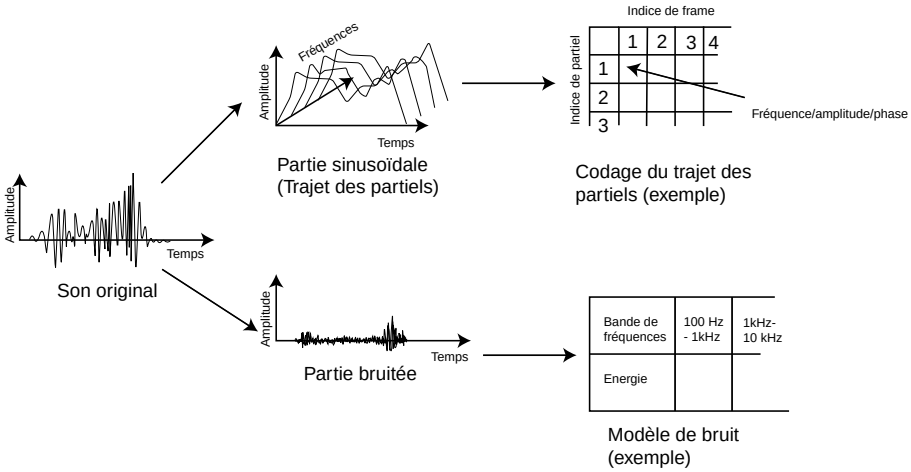


FIG. 1 – Extraction des paramètres d'un modèle sinusoïdal

et à extraire dans chaque fenêtre les pics du spectre, ce qui donne à chaque pas temporel la fréquence, la phase et l'amplitude des partiels. La partie bruit est alors calculé est soustrayant au signal original une somme des sinusoïdes ayant les propriétés précédemment extraites.
Pour réduire la taille de la représentation, on peut alors choisir que de ne garder que les n premiers partiels, ce qui revient à effectuer un filtrage passe-bas, forcément audible.

2 Energie, puissance, valeur RMS, dB SPL : rappels théoriques

Nous allons dans ce paragraphe rappeler quelques définitions usuelles (physiques) de quelques grandeurs utiles au calcul de desripteurs audio (énergie, puissance, valeur RMS, décibels SPL), dans le cas de signaux en temps continus. Nous rappelons ensuite ces définitions pour le cas de signaux en temps discret. Finalement nous proposons notre propre convention de normalisation des calculs pour un ensemble de descripteurs couramment utilisés (spectre, énergie totale, énergie dans des bandes de fréquences, puissance moyenne totale et valeur RMS en dB), et nous appliquons ces définitions sur un exemple.

2.1 Calculs pour les signaux analogiques (en temps continu)

Quelques rappels théoriques pour mettre au point les définitions des quantités usuelles.

2.1.1 Formules dans le domaine temporel

Soit un signal x(t)à support borné ou non dans $\mathbb{R}$ (au hasard un sinus d'amplitude 1 et de fréquence 1000 Hz, défini de t=$-\infty$ à $+\infty$). Sa transformée de Fourier se note X(f).

Energie L'énergie (ou travail) fournie par un signal au cours du temps a pour unité S.I. le *Joule* (J). L'énergie totale du signal entre t=$-\infty$ à $+\infty$ se calcule par la formule 1a, l'énergie fournie à l'instant t par l'équation 1b.

$$E_x = \int_{t=-\infty}^{+\infty} |x(T)|^2 dT \tag{1a}$$

$$E_x(t) = \int_{t=-\infty}^{t} |x(T)|^2 dT \tag{1b}$$

Il en résulte qu'en signal pérdiodique à support non borné (infini) a une énergie infinie.

Puissance La puissance d'un signal représente le débit d'énergie (quantité d'énergie par unité de temps) fournie par le signal. Son unité S.I. est le *Watt* (W). La puissance instantanée délivrée par le signal à l'instant t est définie par l'équation 2a. La puissance moyenne dans l'intervalle $[t_1; t_2]$ est définie par l'équation 2b, et la puissance moyenne sur toute la durée du signal par l'équation 2c.

$$P_x(t) = \frac{dE(t)}{dt} \tag{2a}$$

$$\bar{P}_x(t_1; t_2) = \frac{1}{t_2 - t_1} \int_{t=t_1}^{t_2} |x(t)|^2 dt \tag{2b}$$

$$\bar{P}_x = \lim_{T \to \infty} \frac{1}{T} \int_{t=-T/2}^{T/2} |x(t)|^2 dt \tag{2c}$$

Il en résulte qu'un signal infini d'énergie totale finie a une puissance moyenne totale nulle. Réciproquement un signal infini qui a une puissance moyenne sur toute sa durée non nulle à une énergie nulle.
Pour un signal périodique de période T_0, la puissance totale moyenne peut se calculer sur une seule période. L'équation 2c devient donc l'équation 3

$$\bar{P}_x = \frac{1}{T_0} \int_{t=0}^{T_0} |x(t)|^2 dt \tag{3}$$

Pour les signaux non permanents (non infinis), il est plus judicieux de ne parler que de l'énergie, car la puissance moyenne totale est forcément nulle (énergie finie sur une durée infinie).

Valeur RMS La valeur RMS d'un signal est définie dans l'intervalle [t_1 ;t_2] par l'équation 4. Elle s'exprime dans l'unité de x, avec $_{RMS}$ en indice (e.g. si x représente la pression en Pa, L_{RMS} est en Pa_{RMS}).

$$L_{RMS} = \sqrt{\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} |x(t)|^2 dt} \tag{4}$$

Pour un sinus d'amplitude A, la valeur RMS est égale à $A/\sqrt{2}$ (voir figure 2).

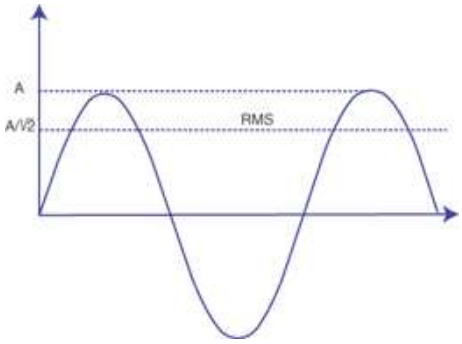


FIG. 2 – Valeur RMS d'un sinus d'amplitude A

dB SPL Les dB SPL (Sound Pressure Level) sont des décibels référencés par rapport à un signal sinsuoïdal de fréquence 1000 Hz de $2e^{-5}$ Pa_{RMS} (seuil d'audition à 1000 Hz). Ils se calculent donc à l'aide de la formule 5.

$$L_{dB} = 20log_{10}(\frac{L_{RMS}}{2e^{-5}}) \tag{5}$$

94 dB SPL correspondent à 1 Pa_{RMS}.

2.1.2 Formules dans le domaine fréquentiel

Relation de Parseval La relation qui permet de passer du domaine temporel au domaine fréquentiel est la relation de Parseval (équation 6).

$$\int_{t=-\infty}^{+\infty} |x(t)|^2 dt = \int_{f=-\infty}^{+\infty} |X(f)|^2 df \tag{6}$$

Energie On peut donc calculer l'énergie dans le domaine fréquentiel à l'aide de la relation 7

$$E_x = \int_{f=-\infty}^{+\infty} |X(f)|^2 df = \int_{f=-\infty}^{+\infty} S_{E_x}(f) df \tag{7}$$

La quantité $S_{E_x}(f) = |X(f)|^2$ est donc la *densité spectrale d'énergie*.

Valeur RMS et dB SPL De la même manière on peut donc exprimer la valeur RMS d'un signal entre t_1 et t_2 (attention dans cette formule $X_{t_1;t_2}(f)$ est la transformée de Fourier du signal fenêtré entre t_1 et t_2) par la relation 8.

$$L_{RMS}(t_1, t_2) = \sqrt{\frac{1}{t_2 - t_1} \int_{f=-\infty}^{+\infty} |X_{t_1;t_2}(f)|^2 df} \tag{8}$$

Puissance Pour la puissance, curieusement, les choses ne sont pas si simples. Il faut appliquer le théorème de Wiener-Kintchine (équation 9).

$$\lim_{t\to\infty} \frac{1}{T} \int_{t=-T/2}^{T/2} |x(t)|^2 dt = \int_{t=-\infty}^{\infty} S_{P_x}(f) df \tag{9}$$

Dans cette équation $S_{P_x}(f)$ est la *densité spectrale de puissance*. Elle se calcule par l'équation 10a, dans laquelle $r_{xx}(\tau)$ est la fonction d'autocorrélation du signal x(t).

$$S_{P_x}(f) = \int_{\tau=-\infty}^{+\infty} r_{xx}(\tau) e^{-j2\pi f\tau} d\tau \tag{10a}$$

$$r_{xx}(\tau) = \int_{t=-\infty}^{+\infty} x(t-\tau)x(t) dt \tag{10b}$$

La densité spectrale de puissance se définit donc comme la transformée de Fourier de la fonction d'autocorrélation du signal. Cette fonction se calcule assez mal pour des signaux échantillonnés. Tout le jeu est alors de trouver un estimateur de $P(f)$ (voir les paragraphes suivants).

2.2 Calculs pour les signaux numériques (temps discret)

Ce qui nous intéresse plus, ce sont ces formules exprimées pour des signaux numérisés à une fréquence d'échantillonnage F_e (voir figure 3). Le signal x(t) numérisé devient le signal x(n). Sa transformée de Fourier discrète (rapide ou non) devient $X_{DFT}(k)$ (attention non normalisée, il faut se souvenir que $X(k*\Delta f) = \frac{X_{DFT}(k)}{N}$: condition de normalisation).

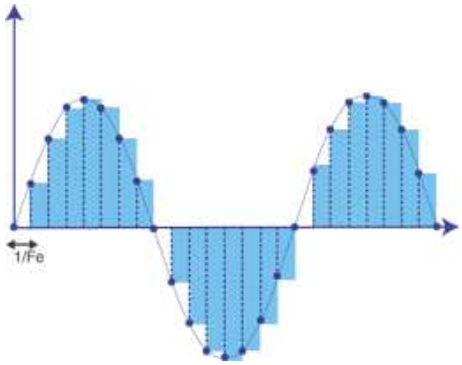


FIG. 3 – Signal numérisé à la fréquence F_e

2.2.1 Formules dans le domaine temporel

Energie La numérisation des équations 1a et 1b donnent respectivement les équations 11a et 11b.

$$E_x = \frac{1}{Fe} \sum_{N=-\infty}^{+\infty} |x(N)|^2 \tag{11a}$$

$$E_x(n) = \frac{1}{Fe} \sum_{N=-\infty}^{n} |x(N)|^2 \tag{11b}$$

Puissance Les équations 2a 2b, et 2c deviennent (je ne suis pas du tout sur de moi sur ce coup-là) les équations 12a 12b, et 12c.

$$P_x(n) = \frac{E_x(n+1) - E_x(n)}{T_e} \tag{12a}$$

$$\bar{P}_x(n_1; n_2) = \frac{1}{(n_2 - n_1)T_e} \sum_{n=n_1}^{n_2-1} |x(n)|^2 Te = \frac{1}{n_2 - n_1} \sum_{n=n_1}^{n_2-1} |x(n)|^2 \tag{12b}$$

$$\bar{P}_x = \lim_{2N\to\infty} \frac{1}{2N} \sum_{n=-N}^{N-1} |x(n)|^2 \tag{12c}$$

Généralement, la puissance est plutôt estimée dans le domaine fréquentiel.

Valeur RMS et dB SPL Pour un signal de N échantillons, l'équation 4 devient l'équation 13

$$L_{RMS} = \sqrt{\frac{F_e}{N} \sum_{n=0}^{N-1} |x(n)|^2 \frac{1}{Fe}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2} \tag{13}$$

Le niveau en dB SPL se calcule de la même manière.

2.2.2 Formules dans le domaine fréquentiel

x(n) est un signal de N échantillons, et $X_{DFT}(k)$ sa transformée de Fourier discrète sur N points non normalisée (c'est donc par exemple directement le résultat de l'algorithme de FFT).

Energie et relation de Parseval Le point clé est d'exprimer la relation de Parseval (équation 6) dans le domaine fréquentiel. Il faut prendre garde à ce que $X_{DFT}(k)$ représente la transformée de Fourier *non normalisée* (i.e. directement la sortie de l'algorithme de FFT). Comme généralement on ne travaille que sur la moitié du spectre (le spectre pour les fréquences négatives est le complexe conjugué du spectre pour les fréquences positives), on prendra plutôt la deuxième forme de l'équation 14.

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_{DFT}(k)|^2 = \frac{2}{N} \sum_{k=0}^{N/2-1} |X_{DFT}(k)|^2 \tag{14}$$

L'énergie peut alors être calculée à partir des équations 15a et 15b (cette équation est définie par analagie avec l'équation 7, avec $\frac{F_e}{N}$ la version discrète de df.

$$E_x = \frac{2}{NF_e} \sum_{k=0}^{N/2-1} |X_{DFT}(k)|^2 \tag{15a}$$

$$E_x = 2 \sum_{k=0}^{N/2-1} S_{E_x}(k) \frac{Fe}{N} \tag{15b}$$

$$S_{E_x}(k) = \frac{1}{F_e^2} |X_{DFT}(k)|^2 \tag{15c}$$

$S_{E_x}(k)$ est donc la densité spectrale d'énergie.

Puissance Le théorème de Wiener-Kintchine (équation 9) s'écrit pour des signaux numérisés au moyen de l'équation 16, avec $S_{P_x}(k)$ la densité spectrale de puissance (la variable d'intégration devient $\frac{F_e}{N}$).

$$P_x = 2 \sum_{k=0}^{N/2-1} S_{P_x}(k) \frac{F_e}{N} \tag{16}$$

Il n'y a pas de moyen de calculer exactement $S_{P_x}(k)$ (je ne sais pas pourquoi). On définit alors des estimateurs de la densité spectrale de puissance. Le plus simple est le *périodogramme*. Il est définit par l'équation 17a.

$$\hat{S}_{P_x}(k) = \frac{1}{NF_e} |X_{DFT}(k)|^2. \tag{17a}$$

$$P_x \approx 2 \sum_{k=0}^{N/2-1} \hat{S}_{P_x}(k) \frac{F_e}{N} \tag{17b}$$

Cet estimateur est biaisé (il n'est pas égal à la vraie densité spectrale de puissance), et non consistant (la variance de l'erreur ne diminue pas quand N augmente).
Un meilleur estimateur est le *périodogramme de Welch*. Il consiste à fenêtrer le signal temporel, à calculer le périodogramme dans chaque fenêtre (avec une fenêtre de forme donnée, de l'overlap et du zéro-padding) et finalement à moyenner les différents périodogrammes (attention, cependant, il faut introduire des corrections liées à la forme de la fenêtre). C'est un estimateur biaisé, mais consistant (la variance de l'erreur tend vers zéro quand N tend vers l'infini).
Pour illustrer cette propriété, prenons le cas d'un bruit blanc gaussien de moyenne $\mu = 0$ et de variance $\sigma^2 = 1$. Un tel bruit a en théorie une densité spectrale de puissance constante égale à $1/F_e$ (voir par exemple [17] page 69[2]). On calcule la DSP par la méthode du périodiogramme et la méthode du périodogramme de Welch (frames de 256 échantillons, 50 % de recouvrement, fenêtres de Hanning, fft calculée sur N points où N est le nombre d'échantillons total du signal : il y a donc N-256 échantillons de zero-padding). Si on estime la DSP sur N=300 points, on obtient la figure 4. On voit sur cette figure que les deux méthodes aboutissent à des estimateurs qui oscillent autour de la valeur théorique avec la même amplitude (l'écart-type de l'erreur est de $2.45e^{-5}$ pour le périodogramme, et de $1.80e^{-5}$ pour le périodogramme de Welch). Néanmoins dans ces deux cas, intéger la DSP permet de retrouver la valeur théorique de la puissance moyenne.
Quand on fait les mêmes calculs pour 10000 points, on s'aperçoit que pour le périodogramme la variance de l'erreure reste du même ordre de grandeur, alors qu'elle diminue considérablement pour le périodogramme de Welch (l'écart-type de l'erreur est de $2.30e^{-5}$ pour le périodogramme, et de $8.25e^{-7}$ pour le périodogramme de Welch).

[2] Attention, on dit bien que c'est la *densité spectrale de puissance* qui est plate. On ne dit rien sur la transformée de Fourier.

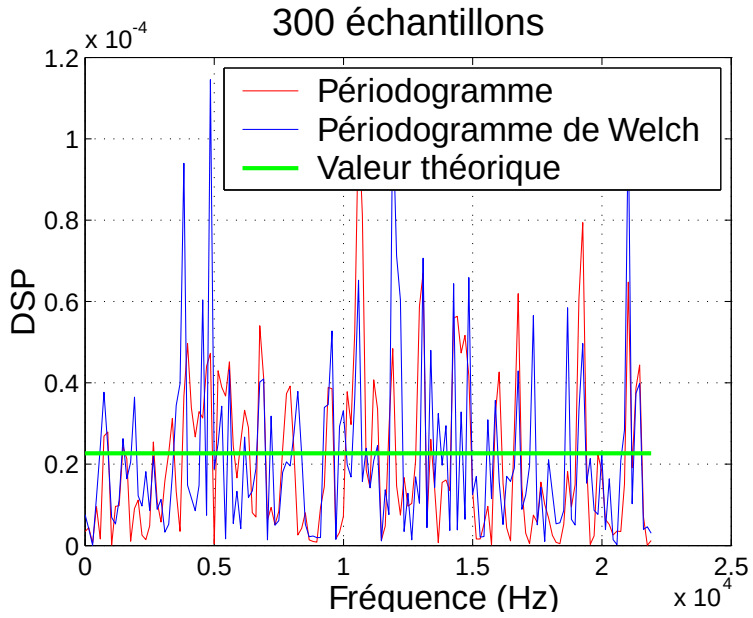


FIG. 4 – Densité de spectrale théorique, estimée sur 300 points par la méthode du périodogramme, par la méthode du périodogramme de Welch, pour un bruit blanc gaussien de moyenne nulle et de variance 1

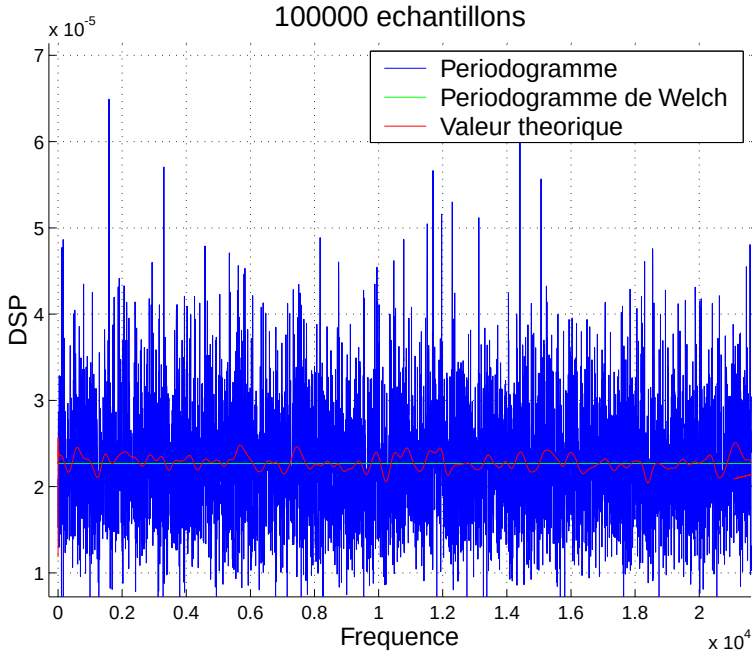


FIG. 5 – Densité de spectrale théorique, estimée sur 100000 points par la méthode du périodogramme, par la méthode du périodogramme de Welch, pour un bruit blanc gaussien de moyenne nulle et de variance 1

Valeur RMS et dB SPL La valeur RMS se calcule donc à l'aide de l'équation 18.

$$L_{RMS} = \sqrt{\frac{2}{N^2} \sum_{k=0}^{N/2-1} |X_{DFT}(k)|^2} \tag{18}$$

Le niveau en dB SPL se calcule de la manière usuelle.

2.3 Propositions de conventions de calcul

Après ces petites mises au point, définissons ce dont nous avons besoin : le **spectre**, l'**énergie totale**, l'**énergie dans des bandes de fréquence**, la **puissance moyenne totale** et la **valeur RMS en dB**. La puissance ne nous sert à rien, et la calcul en est problématique.

2.3.1 Normalisation et cohérence des unités

L'échelle de x(n) n'a pas dans notre cas de signification. Elle est reliée à la pression du signal acoustique diffusé ou mesuré (suivant le cas) par une constante multiplicative k (efficacité ou sensibilité de la chaine de mesure, y compris l'atténuation de propagation) : $x(n) = kP(n)$, qu'on ignore le plus souvent. Ça n'a donc pas vraiment de sens de définir des unités pour l'énergie, ni de référencer les dB par rapport à un sinus de fréquence 1000 Hz à $2e^{-5}\ Pa_{RMS}$.
On propose de prendre comme référence un sinus à 1000 Hz d'amplitude 1. Ce signal a comme valeur RMS $1/\sqrt{2}$. Les dB référencés par rapport à cette valeur seront notés dB_0. On peut montrer par ailleurs que l'énergie de ce signal tend vers T/2 (T est la durée du signal) pour $T \gg T_0$ avec $T_0 = 1/1000$ la période du sinus, et la puissance de ce signal est égale à 1/2 (voir annexe). L'énergie d'un signal ne sera pas normalisée (l'énergie du signal de référence dépend de sa durée, on ne peut pas normaliser par rapport à elle, ou alors ça revient à calculer la puissance moyenne !). La puissance totale sera par contre normalisée par rapport à 1/2.

2.3.2 Conservation d'énergie et énergie dans une bande de fréquence

Spectres en dB Le spectre d'amplitude peut-être représenté en dB dans ce cas, on utilise la formule en $20log_{10}$.

Conservation de l'énergie et énergie par bande Quand on calcule l'énergie dans des bandes de fréquence, il faut s'assurer qu'au final, la somme des énergies dans toutes les bandes donne bien l'énergie totale du signal. Donc l'énergie dans une bande i de bornes $B_{li}(f)$ et $B_{hi}(f)$ s'écrit :

$$E_i = \frac{2}{NF_e} \sum_{k\Delta f \geq B_{li}(f)}^{k\Delta f < B_{hi}(f)} |X_{DFT}(k)|^2$$

De cette manière, on a bien $\sum_i E_i = E_x$. Cela revient en fait à calculer une nouvelle densité spectrale d'énergie, avec $\Delta f_i = B_{hi}(f) - B_{li}(f)$ la largeur de la bande i :

$$S_{E_i} = \frac{1}{2}\frac{E_i}{\Delta f_i}$$

On vérifie qu'on a bien (cette équation est à mettre en regard de l'équation 15b) :

$$E_x = 2\Delta f_i \sum_i S_{E_i}$$

Néanmoins, il est sans doute préférable de conserver l'énergie dans les bandes E_i, plutôt que la densité spectrale d'énergie S_{E_i} par bande. En effet, comme souvent les bandes sont de tailles non constantes, il faudrait aussi connaître la taille de chaque bande Δf_i pour retrouver l'énergie dans chaque bande.

2.3.3 Résumé des formules normalisées

Soit un signal x(n) de durée N échantillons à la fréquence d'échantillonnage F_e.

2.4 Exemples

Ces exemples doivent servir à vérifier les calculs qu'on implémente.

- Un sinus de fréquence 1000 Hz d'amplitude 1 échantillonnée à 44.1 kHz (voir la figure 6) de durée 1 s
- Le même sinus d'amplitude 0.5
- Un bruit blanc entre -1 et 1, de durée 1s échantillonné à 44.1 kHz
- Un bruit gaussien de 1s échantillonné à 44.1 kHz, de moyenne nulle et d'écart-type 1

Grandeur	Symbole	Calcul en temporel	Calcul en fréquentiel
Signal	$x(n)$		
Transforméee de Fourier discrète	$X_{DFT}(k)$		
Spectre	$X(k)$		$X_{DFT}(k)/N$
Spectre d'amplitude	$\|X(k)\|$		
Spectre de phase	$\angle X(k)$		
Puissance de référence	P_0	$1/2$	
Puissance moyenne totale	$\bar{P}_x$	$\frac{1}{N}\sum_{n=0}^{N-1}\|x(n)\|^2$	$2\sum_{k=0}^{N/2-1}\|X(k)\|^2$
Puissance moyenne totale normalisée	$\bar{P}_{x0}$	$\bar{P}_x/P_0$	
Energie totale	E_x	$\sum_{n=0}^{N-1}\|x(n)\|^2/F_e$	$2\frac{N}{F_e}\sum_{k=0}^{N/2-1}\|X(k)\|^2$
Energie dans la bande i	E_i		$2\frac{N}{F_e}\sum_{k\Delta f<B_{hi}(f)}^{k\Delta f\geq B_{li}(f)}\|X(k)\|^2$
Valeur RMS	L_{RMS}	$\sqrt{1/N\sum_{n=0}^{N-1}\|x(n)\|^2}$	$\sqrt{2\sum_{k=0}^{N/2-1}\|X(k)\|^2}$
Niveau en dB_0	L_{dB_0}	$20log_{10}(L_{RMS}*\sqrt{2})$	

TAB. 1 – Résumé des formules normalisées

	Sinus@1kHz	Sinus@1kHz/2	Bruit blanc entre -1 et 1	Bruit gaussien μ=0 σ=1
Energie totale	0.500	0.125	0.333	0.983
Puissance totale	0.500	0.125	0.333	0.983
Puissance totale normalisée	1.00	0.250	0.666	1.967
Valeur RMS	0.707	0.353	0.577	0.992
Niveau en dB_0	0.000	-6.000	-1.761	2.938
Figure	fi gure 6		fi gure 7	fi gure 8

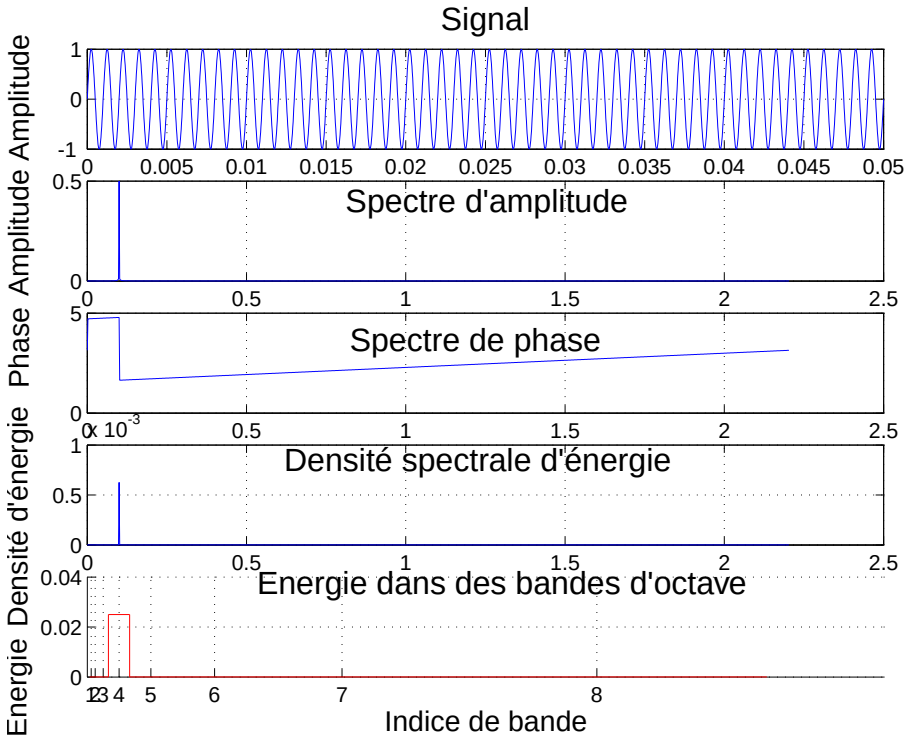


FIG. 6 – Spectres et densités spectrale d'énergie normalisée d'un sinus d'amplitude 1 et de fréquence 1000 Hz

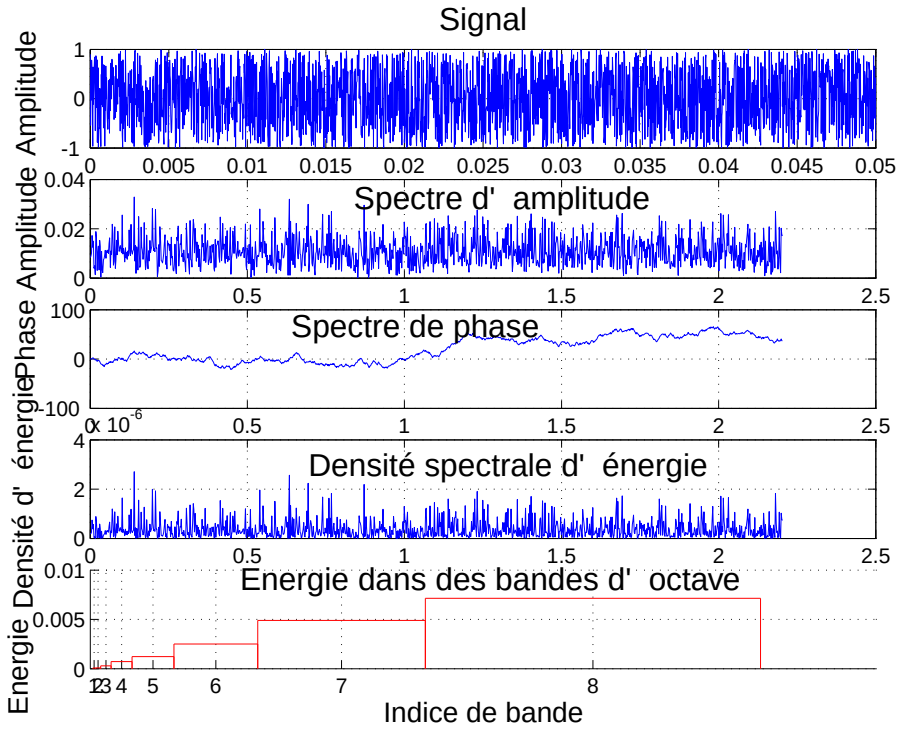


FIG. 7 – Spectres et densité spectrale d'énergie normalisée d'un bruit blanc entre -1 et 1

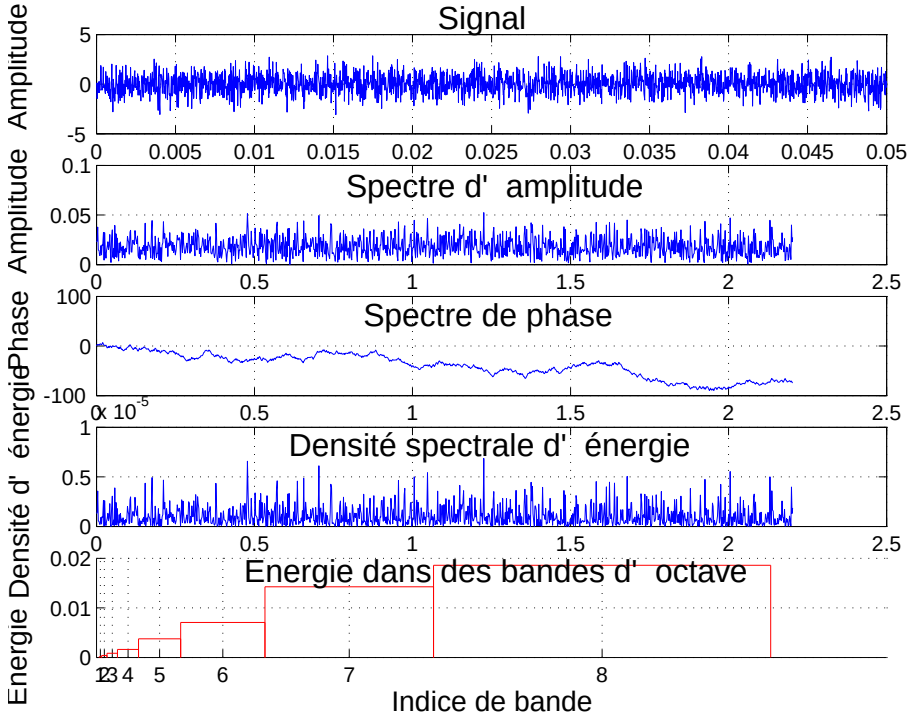


FIG. 8 – Spectres et densité spectrale d'énergie normalisée d'un bruit gaussien de moyenne nulle de d'écart-type 1

3 Matlab Sound Processing Library (SPL). Version 1.15. Spécification des fonctions

La librairie Matlab SPL contient des fonctions qui permettent de calculer les descripteurs et de les inclure dans un fichier au format spécifique (.sig). Cette librairie contient donc également les fonctions d'écriture et de lecture des fichiers. Les trois fonction principales sont *Sigwrite.m*, *SigLoad.m* et *SigAdd.m*, qui, calculent respectivement les descripteurs, les stockent dans un fichier .sig, lisent un fichier .sig, et ajoutent un descripteur à un fichier .sig préexistant. Ces fonctions sont définies dans le paragraphe 3.1. Quelques fonctions de test sont décrites dans le paragraph 3.2. Le détail des fonctions qui calculent les descripteurs sont décrits dans le paragraphe 3.3.

3.1 Fonctions générales

.

SigWrite.m

Description :
Computes all the descriptors and write a .sig file
Syntax :
out=SigWrite(in,filename,SR,FrameSize,WindowType,HopSize) ;
Inputs :

- in : input signal in a vector (FrameSize samples)
- filename : name of the .sig file to be written. If filename.wav does not exist, it is created. If in is an empty vector, filename.wav is read instead.
- SR : Sampling rate
- FrameSize : size of the frame, used for the segmentation
- WindowType : type of windowing : → 'Rectangular' → 'Hamming' → 'Hanning'
- HopSize : size of the overlap in samples

Output :

- out.nchan : number of channels
- out.srate : sampling rate
- out.nsamp : total size of the original signal
- out.nfram : number of frames
- out.fsize : size of the frames samples
- out.hsize : hop size
- out.wtype : window type
- out.corr : windowing correction
- out.nfft : discrete normalized Fourier transform (out.nfram*(out.fsize/2+1) values)
- out.dct : discrete cosine transform (out.nfram*out.fsize values)
- out.aspe : amplitude spectrum (out.nfram*out.fsize/2 values)
- out.pspe : power spectrum (out.nfram*out.fsize/2 values)
- out.ceps : cepstrum (out.nframe*out.fsize values)
- out.mfcc : mel frequency cepstral coefficients (out.nframe*16 values)
- out.bark : bark power (out.nframe*32 values)
- out.tona : tonlity index (out.nframe values)
- out.mask : masking threshold (out.nframe values)
- out.lev0 : RMS level in db0 (out.nframe values)
- out.ava : average amplitude (out.nframe values)
- out.spce : spectral centroid (out.nframe values)
- out.sigl : signal (out.nsamp values)

The sigwrite procedure is summarized in figure 9.

SigLoad.m

Description :
Read a .sig file.
Syntax :
out=SigLoad(filename) ;
out is the same format as described before.

SigAdd.m

Description :
Add a new data chunk to an existing .sig file
Syntax :
[]=SigAdd(filename,TAG,datachunk) ;
Input :

- filename : name of the .sig file
- TAG : tag of the data chunk (4 characters string)
- datachunk : data to be added to the file. datachunk m ay be either a vector or a matrix.

3.2 Fonctions de test

TestSigReadWrite.m

Description :
Synthesize a 1000 Hz pure tone, write the associated .sig file, and reread it. This functions displays the correlation coefficients between read and writeen descriptors. All correlation coefficients must be 1.

TestBasicSignals.m

Description :
Synthesize a 1000 Hz pure tone, and a gaussian white noise (average = 0, standard deviation is 1), write the associated .sig file, and reread it. This functions computes the mean power of the total signals by several means. All computed powers must be equal to the theoretical power.

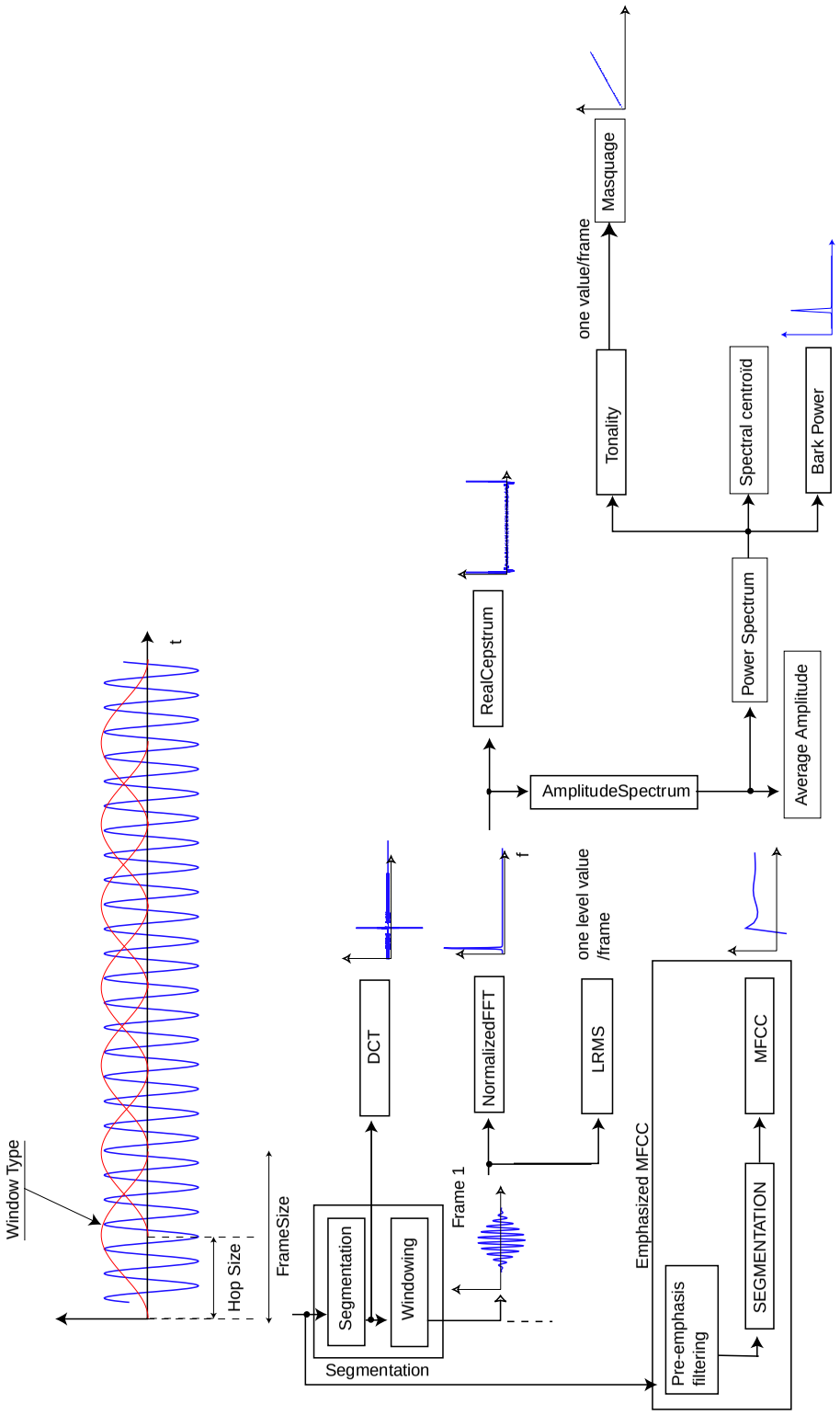


FIG. 9 – Schéma de principe du calcul des descripteurs.

3.3 Fonctions de caluls des descripteurs

AverageAmplitude.m

Description :
Computes the average amplitude of the amlitude spectrum.
Syntax :
A= AverageAmplitude(Amplitude)
Input :
- Amplitude : Amplitude spectrum (must be out.nframe/2 long)

BarkPower.m

Description :
This function computes the power of the signal within one-bark long windowed.
Syntax :
out=BarkPower(PS,SR,FrameSize) ;
Input :
- PS : PowerSpectrum (FrameSize/2 values)
- SR : Sampling Rate
- FrameSize : size of the frame

Output :
- out : power of the signal in each Bark band (26 values)

Procedure : The power spectrum is averaged over one-bark wide rectangular windowed without any overlap. The double of the sum of the individual band powers give the total power averaged over the frame of the signal.

DCT.m

Computes the classical discrete Cosinus transform. See the dct.m matlab function in the signal processing toolbox.

EMfcc.m

Description :
This function computes the emphasized Mel frequency cepstral coefficients. This implementation is inspired by the mfcc.m function that may be found within the Malcom Slaney's auditory toolbox.
Syntax :
[mfcc]=mfccSig(input,SamplingRate,FrameSize,HopSize,WindowType,Ncc,pre,MethMel) ;
Input :
- input : input signal (not segmented)
- SamplingRate : Sampling rate in Hz (default 44100)
- FrameSize : size of the Frame in samples (default 1024)
- HopSize : size of the overlap, in samples (default 0)
- WindowType : type of windowing → 'Rectangular' → 'Hamming' → 'Hanning' (default 'Rectangular')

OPTIONS
- Ncc : number of kept coefficients (default : 13)
- pre : pre-emphasizing filtering → 1 yes → 0 no (default : no)

Output
- mfcc : mel frequency cepstral coefficients (Ncc*Nframe values)

Procedure : The input signal is first low-pass filtered, then segmented and windowed. For each frame, the magnitude of the discrete Fourier transform is computed and averaged within 1-mel long triangular-weighted windows. Finally the logarithm of the averaged magnitude spectrum is transformed by means of a discrete cosinus transform. This procedure is summarized by figure 10.

Mfcc.m

Description :
Same function as Emfcc, without the pre-emphasis filltering. Hence, the input is directly the magintude spectrum.
Syntax :

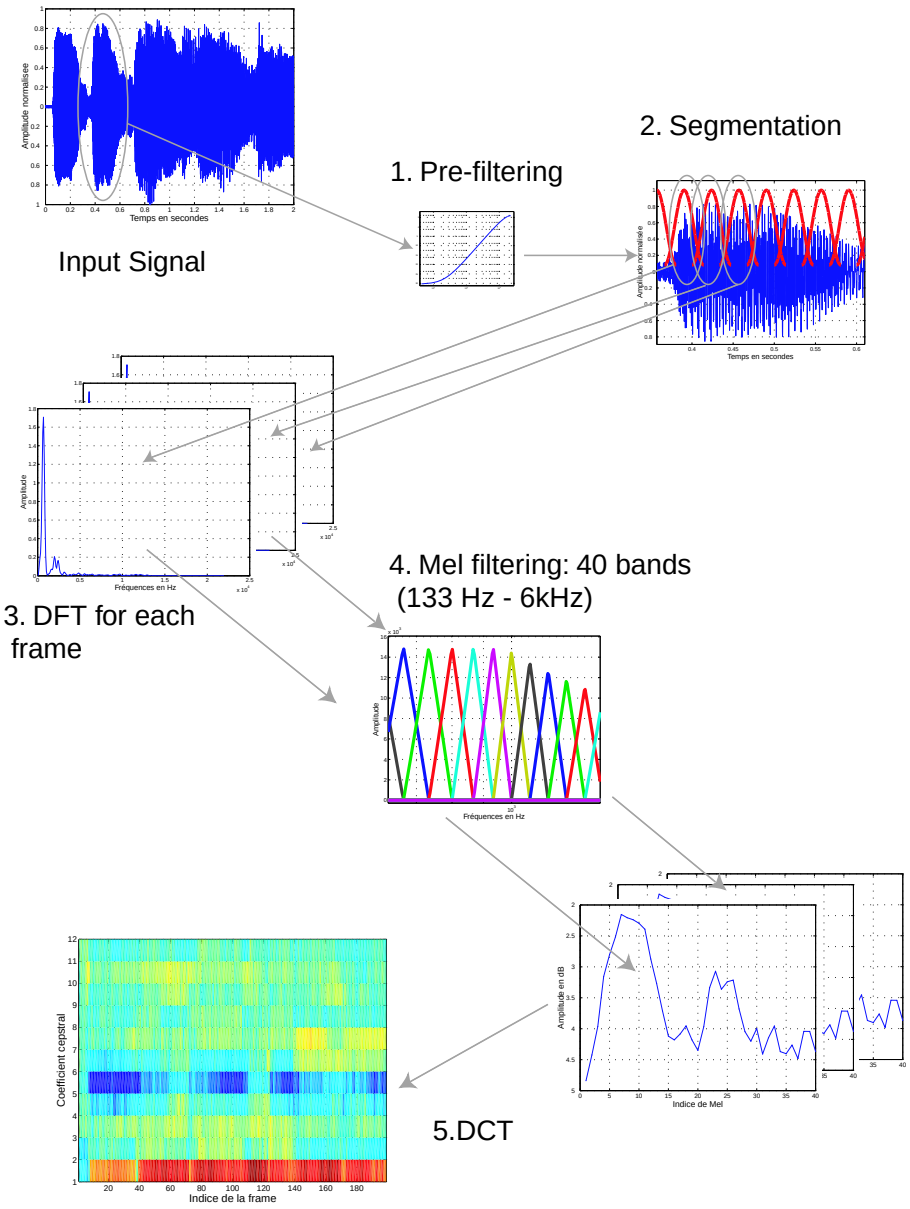


FIG. 10 – Schéma de principe du calcul des mfcc.

[mfcc] = Mfcc(FftMag,SR,FrameSize,Ncc) ;
Input :
- FftMag : magnitude of the FFT (not normalized, FrameSize/2+1 values)
- SR : Sampling Rate
- FrameSize : size of the original signal
- Ncc : number of kept coefficients

Output :
- mfcc : mel-frequency cepstral coefficients (Ncc values)

LRMS.m

Description :
This function computes the RMS level of the input segments, in decibels normalized to $1/\sqrt{2}$ (labeled dB_0, so that a pure sine one of amplitude 1 will have a RMS level of 0 dB_0).
Syntax :
L=LRMS(in,alpha) ;
Input :
- in : input signal in a vector
- alpha : windowing correction (default : 1)

Output :
- L : RMS level normalized to $1e^{-5}/\sqrt{2}$

Procedure :
The level is computed according to the relations :

$$L_{RMS} = \sqrt{\frac{1}{N\alpha}\sum |x|^2} \tag{19}$$

$$L_{dB} = 20log_{10}\frac{L_{RMS}}{1e^{-5}/\sqrt{2}} \tag{20}$$

Masquage.m

Description :
This function computes the masking threshold for each specified frequency band.
Syntax :
Th=Masquage(T,SR,FrameSize,IntegSpec)
Input :
- T : Tonality index
- SR : SamplingRate (default 1024)

OPTIONS
- IntegSpec : Spectral integration 'quatre' : 4 bands(see [25])/ 'bark' : 26 one-bark wide bands

Output :
- Th : Masking thresholds, per one-Bark width band (26 values)

Procedure :
The masking thresholds are computed according the relation : thr=(14.5+B).*T+5.5*(1-T), where B is the band index and T the tonality index.

NormalizedFFT

Description :
This function computes the normalized discrete Fourier transform.
Syntax :
[ff]=NFFT(in,FrameSize,alpha)
Input :
- in : original signal
- SR : sample rate
- FrameSize : frame size, in samples

- alpha : window correction

Output :

- ff : normalized Fourier transform (FrameSize/2 +1 complex values)

Procedure :

The fast Fourier transform is first computed by means of the classical FFT algorithm. It is then normalized according to the relation : $X = FFT/\alpha N$. Finally, the upper half of the vector (corresponding to the negative frequencies) is remove. Hence the size of ff is FrameSize/2 + 1. It should be noted that ff is a vector of complex numbers.

RealCepstrum.m

Description :

This function computes the real part of the cepstrum, as defined [16] page 507 for instance.

Syntax :

c=RealCepstrum(TF,FrameSize,alpha) ;

Input :

- TF : Fourier transform (normalized FFT of the signal). Must be FrameSize/2 +1 long
- FrameSize : frame size (default : 1024)
- alpha : window correction (default : 1)

Output :

- c : real part of the cepstrum (FrameSize values)

Procedure :

The fast Fourier transform is first computed by de-normalizing the Fourier transform, and wrapping it to the negative frequencies. The cesptrum is then computed after the formula : $c = ifft(ln|fft|)$, where ifft is the inverse fast Fourier Transform algorithm.

Segmentation.m

Description :

This function segments the original into FrameSize-samples frames, weighted by a WindowType window, with a HopSize-samples overlap.

Syntax :

[x,xNW,NFrame,alpha]=Segmentation(in,FrameSize,HopSize,WindowType) ;

Input :

- in : original signal
- FrameSize : frame size, in samples
- HopSize : overlap size, in samples
- WindowType : type of window → 'Rectangular' → 'Hamming' → 'Hanning'

Output :

- x : FrameSize*NFrame matrix of windowed frames
- x : FrameSize*NFrame matrix of not-windowed frames
- NFrame : total number of frames
- alpha : window correction

Procedure :

The signal is first zero-padded, in order to have its length an integer multiple of FrameSize - HopSize. Then the frames are windowed (or not) and segmented. The correction coefficient alpha is used to weigh the Fourier transform (see next paragraph). It is computed according to equation : alpha : $\alpha = \sqrt{1/N \sum w(n)^2}$, where N is the total length of the signal in samples, and w(n) the sampled window.

SpectralCentroid.m

Description :

Computes the spectral centroïd of the spectrum (power-weighted frequency average)

Syntax :

SC=SpectralCentroid(PS,SR,FrameSize) ;

Input :

- PS : Power spectrum (FrameSize/2 values)
- SR : Sampling Rate

- FrameSize : size of the frame

Procedure :
The spectral centroïd is computed according to the relation :

$$SC = \frac{\sum f * PS}{\sum PS} \tag{21}$$

Tonality.m

Description :
This function computes the tonality of the frame
Syntax :
ton=Tonality(PS,SR,FrameSize)
Input :

- PS : Power spectrum (FrameSize/2 values)
- SR : SamplingRate (default 44100)
- FrameSize : Size of the frame (default 1024).

o output

- ton : tonality (Nframe values)

Procedure :
The spectral flatness measure is first computed according to equation :

$$SFM = 10log_{10}\frac{\mu_g}{\mu_a} \tag{22}$$

Where μ_g and μ_a are respectively the geometric and the arithmetic means of the squared magnitude spectrum. The tonality index is then computed after T= min([-SFM/60 1]).

4 Structure d'un fichier de description ".sig"

Version 0.001 octobre 2004.
Cette partie décrit le format des fichiers ".sig". Ce format est utilisé pour stocker des descripteurs associés à un fichier audio. Il est conçu de manière à être extensible, basé sur une structure en "chunks" pour pouvoir être parcouru facilement. Il a aussi des extensions pour être indexable.

4.1 Description du format

Les données sont groupées dans des "chunks" séparables. Chaque chunk contient un en-tête et les données. L'en-tête décrit le type de données et la taille des donnéees. Cette organisation du fichier permet à un programme qui ne connaîtraît pas un type de donnée de le sauter pour passer à la suite du fichier.

4.2 En-tête du fi chier .sig

Un fichier .sig commence par un en-tête qui décrit le signal audio.

Bytes	Name	Type
4	Header identification string = "SIG "	string
4	Version string = "100 "	string
80	Wav filename	string
4	Crc of the wav file	integer
4	Number of channel	integer
4	Sampling rate	integer
4	Number of samples	integer
4	Number of frames	integer
4	Minimum intensity of the signal	float
4	Maximum intensity of the signal	float

4.3 Chunks de descripteurs

Le reste du fichier est organisé sous forme de chunks. Chaque chunck commence avec 4 octets d'identification and la longueur du chunk. Les identificateurs du chunk sont :

Bytes	name	type
4	Identification string	string
4	Length of the chunk	integer

IDENTIFICATION STRING	Description
'SIGL'	Signal
'FFT '	Fourier transform
'DCT '	Discrete cosine transform
'SPEC'	Spectrum
'CEPS'	Cepstrum
'MFCC'	Mel frequency cepstral coefficients
'BARK'	One-Bark wide band power spectrum
'TONA'	Tonality
'MASK'	Masking threshold
'WAVL'	Wavelets
'SPCE'	Spectral centroid
'KEYW'	Keyword

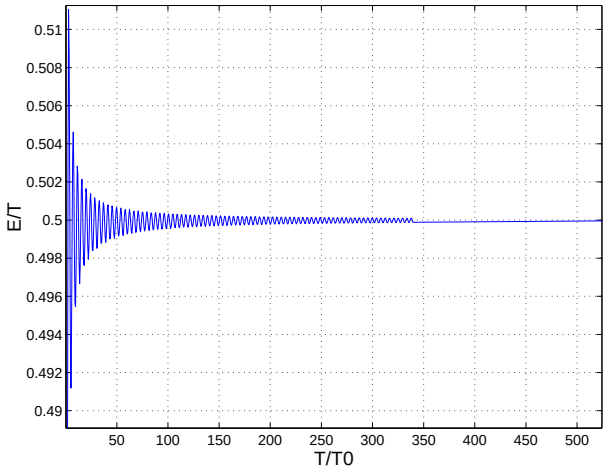


FIG. 11 – Evolution de l'énergie (rapportée à sa durée) fournie par un sinus à 1000 Hz en fonction de sa durée

Annexe : Calcul de l'énergie et la puissance d'un sinus

L'énergie d'un sinus de durée T $x(t) = sin(2\pi f_0 t)$ se calcule à l'aide de la formule suivante (on définit l'origine des temps de telle manière à ce que le déphasage soit nul) :

$$E_x = \int_{t=0}^{T} sin^2(2\pi f_0 t)dt$$

On s'aide de la propriété :

$$\int_{t=0}^{2\pi} sin^2(x)dx = \pi$$

Avec le changement de variable $x = 2\pi f_0 t$, $dx = 2\pi f_0 dt$, on obtient :

$$\int_{t=0}^{T_0} sin^2(2\pi f_0 t)dt = T_0/2$$

Si le signal a une durée $T = kT_0 + \epsilon$ avec $\epsilon \ll kT_0$, on peut écrire :

$$\int_{t=0}^{T} sin^2(2\pi f_0 t)dt = \int_{t=0}^{kT_0} sin^2(2\pi f_0 t)dt + \int_{t=kT_0}^{kT_0+\epsilon} sin^2(2\pi f_0 t)dt$$

La première intégrale vaut $kT_0/2$, et la deuxième est comprise entre 0 et $T_0/2$. On peut donc écrire, avec α entre 0 et 1 :

$$\int_{t=0}^{T} sin^2(2\pi f_0 t)dt = (k+\alpha)T0/2$$

Comme par définition $\alpha \ll k$, on a :

$$E_x = T/2$$

La figure 11 représente l'évolution de l'énergie divisée par la durée du signal, en fonction de la durée du signal exprimée en périodes fondamentales, pour un sinus à la fréquence 1000 Hz. On constate que cette énergie converge vers $T/2$ à partir de quelques centaines de périodes fondamentales.

Du coup, on a d'après l'équation 2b :

$$\bar{P}_x(0;T) = \frac{1}{T}\int_{t=0}^{T} |x(t)|^2 dt = \frac{1}{2}$$

Références

[1] Silvia Allegro, Michael Büchler, and Stefan Launer. Automatic sound classification inspired by auditory scene analysis. In *Consistent and Reliable Acoustic Cues for Sound Analysis (CRAC), one-day workshop, Aalborg, Denmark, Sunday September 2nd 2001 (directly before Eurospeech 2001)*, 2001.

[2] Albert S. Bregman. *Auditory Scene Analysis, The perceptual organization of sound.* The MIT Press, second edition, 1999.

[3] Depalle, Garcia, Rodet, Woehrmann, and Perry. *The additive analysis-synthesis package*. Ircam. htpp ://www.ircam.fr/equipes/analyse-synthese/DOCUMENTATIONS/additive.

[4] Myriam Desainte-Catherine and Pierre Hanna. Statistical approach for sound modeling. In *Proceedings of the* 3^{th} *COST-G6 International Conference on Digital Audio Effects DAFX'00*, 2000.

[5] Richard S. Goldhor. Recognition of environmental sounds. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'93)*, volume 1, pages 149–152, Minneapolis, 1993.

[6] Pierre Hanna. Modélisations de bruits, 2000.

[7] William M. Hartmann. *Signals, sound and sensation*. Springer, 1998.

[8] Perfecto Herrera, Xavier Serra, and Geoffroy Peeters. Audio descriptors and descriptors schemes in the context of MPEG-7. In *Procedings of International Computer Music Conference (ICMC99)*, 1999.

[9] Perfecto Herrera, Xavier Serra, and Geoffroy Peeters. A proposal for the description of audio in the context of MPEG-7. In *Proceedings of the CMBI'99 European Workshop on Content-based Multimedia indexing*, 1999.

[10] James M. Kates. Classification of background noises for hearing-aid applications. *Journal of the Acoustical Society of America*, 97(1) :461–470, January 1995.

[11] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval (Music IR 2000), Plymouth Massaschussetts october 23-25 2000*, 2000.

[12] Stephen McAdams and Nicolas Misdariis. Perceptual-based retrieval in large musical sound databases. In P. Lenca, editor, *Proceedings of the Human Centered Processes Conference*, pages 445–450, Brest, France, September 1999.

[13] Nicolas Misdariis, Bennett K. Smith, Daniel Pressnitzer, Patrick Susini, and Stephen McAdams. Validation of multidimensional distance model for perceptual dissimilarities among musical timbres. In *Proceedings of the* 116^{th} *International Congress on Acoustics (ICA) and* 135^{th} *Meeting of the Acoustical Society of America (ASA)*, Seattle, Washington, 20-26 June 1998.

[14] Brian C. J. Moore. *An introduction to the psychology of hearing*. Academic Press, second edition, 1988.

[15] Peter Nordqvist and Arne Leijon. Automatic classification of the telephone listening environment in a hearing aid. TMH-QPSR, Speech, Music and Hearing, KTH, Stockholm, Sweden, 2002.

[16] Alan V. Oppenheim and Ronald W. Schaffer. *Digital Signal Processing*. Prentice Hall International, 1975.

[17] Alan V. Oppenheim, Ronald W. Schaffer, and John R. Buck, editors. *Discrete-time signal processing*. Prentice Hall International, 2^{nd} edition, 1999 (First edition 1989).

[18] Ted Painter and Andreas Spanias. A review of algorithms for perceptual coding of digital audio signals. In *Proceedings of the International Conference on Digital Signal Processing*, pages 179–205, 1997.

[19] Geoffroy Peeters. *Timbre Toolbox Documentation.* Ircam, Equipe Analyse-Synthèse, 2000. Version 1.0.

[20] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Cuidado projet report, Institut de Recherche et de Coordination Acoustique Musique (IRCAM), 2004.

[21] Geoffroy Peeters, Amaury La Burthe, and Xavier Rodet. Toward automatic music audio summary generation from signal analysis. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR 2002)*, 2002.

[22] Geoffroy Peeters and Xavier Rodet. Automatically selecting signal descriptors for sound classification. In *Proceedings of the International Computer Music Conference (ICMC2002)*, 2002.

[23] Vesa Peltonen, Juha Tuomi, and Anssi Klapuri. Computational auditory scene recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'02)*, Orlando, Florida, May 2002.

[24] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition.* Prentice Hall, 1993.

[25] Nicolas Tsingos, Emmanuel Gallo, and Georges Drettakis. Perceptual audio rendering of complex virtual environments. *ACM Transactions on Graphics*, 23(3), 2004.

[26] Barry L. Vercoe, William G. Gardner, and Eric D. Scheirer. Structured audio : Creation, transmission, and rendering of parametric sound representations. In *Proceedings of IEEE*, volume 86, pages 922–939, may 1998.

[27] Erling Wold, Thom Blum, Douglas Keislar, and James Wheaton. Content-based classification, search and retrieval of audio. *IEEE Multimedia*, 3(3) :27–36, September 1996.

[28] Matthew Wright, Amar Chaudhary, Adrian Freed, David Wessel, Xavier Rodet, Dominique Virolle, Rolf Woehrmann, and Xavier Serra. New applications of the Sound Description Interchange Format. In *Proceedings of the International Computer Music Conference (ICMC-98)*, Ann Arbor, MI, 1998.

[29] Tong Zhang and C.-C. Jay Kuo. Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on Speech and Audio Processing*, 9(4) :441–457, May 2001.

[30] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics Facts and Models*. Springer Verlag, 1990.

Table des matières

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803